Choosing Product Space: Lessons from the App Economy*

Mark A. Jamison† Byoungmin Yu‡

January 6, 2026

Click here for the latest version.

Abstract

Firms often choose with whom to compete and how similar or how different their products should be relative to those of their rivals. This paper investigates this issue in the app economy by studying the determinants of mobile app success. We leverage natural language processing and unsupervised machine learning to cluster apps using pairwise cosine similarity, which provides a measure of horizontal differentiation within app categories. We find asymmetric effects of rivalry from first-party apps: Apple's entry into a cluster stimulates sales of third-party apps, whereas Google's entry decreases them. We also find that apps in a cluster that more closely resemble the most popular apps have fewer downloads, whereas greater similarity to average competitors enhances downloads. We further find that more frequent updates increase app downloads. These findings yield important implications for developers in choosing market segments and designing effective differentiation strategies.

Keywords: mobile app stores, cosine similarity, horizontal differentiation, platform entry, unsupervised machine learning.

JEL classification: L25, L21, C55, L13

*We thank Sensor Tower for providing the data for this research, and the Digital Markets Initiative for financial support. All errors are our own.

†Public Utility Research Center and Digital Markets Initiative, Warrington College of Business, University of Florida, Gainesville, Florida, United States. (mark.jamison@warrington.ufl.edu).

‡Public Utility Research Center and Digital Markets Initiative, Warrington College of Business, University of Florida, Gainesville, Florida, United States. (byoungminyu9@gmail.com).

1 Introduction

Situations arise in which firms compete with their suppliers. National brands, such as Coca Cola and Del Monte, often compete with store brands, such as Kroger's *Big K* and *simply Kroger®*. WorldCom's MCI in the US once relied upon the facilities of local telephone companies to compete against them. AT&T and Xfinity provide cable/fiber broadband in Knoxville, Tennessee, relying upon city permits and rights of way, while competing with the city's Knoxville Utilities Board, which provides utility and broadband services.

The downstream rivalry raises strategic issues, such as product differentiation, pricing, and the potential for sabotage. National brands and store brands often differentiate themselves through advertising, quality differences, and distribution, with price differences reflecting differences in customers' beliefs about differences in value. In telecommunications, wholesale pricing of network facilities was a controversial regulatory issue. Ultimately, relying on facilities of local telephone companies was generally unsustainable as a business strategy for new entrants and companies such as MCI were acquired by incumbents, such as Verizon. City-owned broadband providers raise issues of cross-subsidy as they are rarely commercially viable (Yoo et al., 2022).

These issues are also at play in digital markets and take on new dimensions of complexity. Digital platforms have substantially lowered entry costs for downstream software-based products, leading to an expanding supply. This has been particularly true in the case of mobile applications (apps) that are used on platforms provided by Apple and Google. App markets are attractive to digital entrepreneurs; global consumer spending on in-app purchases reached $150 billion in 2024, up from $130 billion in 2023 (Statista, 2025). But app developers are highly reliant on the tools provided by the platforms. Developers on Apple's iOS platform, for example, must use Apple's Xcode to build, debug, and test apps, and App Store Connect for managing submissions, metadata, pricing, and sales. Other tools may be required depending on the app (Apple, 2025a,b,c,d).

App developers appear to find promise in the app markets: The number of apps continues to expand steadily—from 3.4 million in 2020 to 3.8 million in 2023 (Statista, 2024). But as is generally true for product markets, the distribution of supply across apps is highly skewed as shown in Figure 1, approximating a semi-logarithmic distribution of Buzzell (1981). In other words, the mobile app ecosystem exhibits a pronounced short-tail distribution (Jiang et al., 2011; Zhong and Michahelles, 2013), where many apps enter but few 'Superstar' apps dominate the markets. This raises issues of developer strategy, which motivates a fundamental question: What determines the success of third-party apps in mobile platforms?

Previous studies of app markets primarily attribute app performance to observable app-

level characteristics, such as user ratings (Sällberg et al., 2023), rankings (Carare, 2012; Garg and Telang, 2013; Gokgoz et al., 2021), or offering in multiple categories (Lee and Raghu, 2014). While informative, these explanations offer limited guidance on how developers should strategically position their products relative to competitors within the same functional market segment. In particular, the literature lacks systematic evidence on the extent to which horizontal differentiation among apps—i.e., similarity in feature content and functional attributes—shapes competitive outcomes at the within-cluster level.

This paper seeks to fill this gap by developing a feature-based representation of clusters of apps and examines how an app's horizontal and vertical positioning affects its performance. Using natural language processing to encode app feature descriptions, we construct pairwise comparisons of apps to identify market clusters. Within each cluster, horizontal differentiation is captured by an app's cosine similarity to other apps. Vertical differentiation is proxied by the intensity of its update release, which reflects ongoing investment in quality and functionality.

Our empirical analysis yields three core findings. First, we document that when an app's features are less differentiated from those of other apps in the same cluster, it tends to have more app downloads. However, having features similar to highly popular apps in the same cluster has a negative effect on app downloads, indicating that being too similar to the most popular apps creates a competitive disadvantage. The results suggest a differentiation strategy of being similar to clusters of apps—perhaps indicating the presence of high valuations—but dissimilar to the most popular apps, perhaps by introducing novel attributes that distinguish the app from the famous apps. The effects of this strategy are perhaps reflected in other situations: In telecommunications, relying on incumbents' network facilities made differentiation difficult, contributing to that being a largely failed business strategy. In groceries, store brands enter markets like sodas where demand is high, but choose lower price points than national brands, perhaps reflecting a strategy to serve market niches.

Second, increasing the frequency of app updates leads to more installations, consistent with the notion that continual product improvement reinforces vertical positioning advantages. Lastly, this paper accounts for the influence of differences in platforms. In particular we find that Apple's supply of first-party apps tends to reinforce third-party success within the same cluster, while Google's first-party entry generally exerts a negative effect. This asymmetry likely reflects differences in Apple's and Google's platform and app strategies. Apple's apps are generally positioned in clusters characterized by higher average similarity—indicating less horizontal differentiation—whereas Google's apps are typically positioned in more differentiated clusters. Moreover, the average similarity between third-party apps and Apple's own apps is higher than that between third-party apps and Google's, suggesting

that Apple's products are more closely aligned with the features of third-party apps.

These patterns, combined with the main results, may imply that Apple positions its apps closed to third-party apps, perhaps aiming to expand consumer awareness and demand within those clusters (Li and Agarwal, 2017; Foerderer et al., 2018). But it might also be true that Apple's primary interest is in creating an ecosystem that enables it to sell more iPhones and other devices, implying that the app strategy is primarily one of stimulating customer satisfaction with the ecosystem, even if that means promoting success of third-party apps. In contrast, Google's primary interest is likely to be in selling advertising as it provides no devices of its own. Consistent with that, Google places its apps in more numerous, and relatively distinct product spaces compared to Apple, with novel features that users value. This diverts user attention away from rivals, perhaps offsetting any market expansion effect of Google's apps. Our findings are also consistent with the conclusions of Farrell and Katz (2000) that platform's entry into downstream market exerts a competitive squeeze, and those of Belleflamme and Peitz (2019) that Google's apps prompt developers to reoptimize their innovation and investment strategies.

A primary contribution of this paper is to identify determinants of app success in a two-sided platform through the lens of horizontal differentiation of app features. Our approach provides important implications for app developers in choosing market segments and designing effective differentiation strategies. Our results suggest that app developers entering a product space where their apps share a core feature with competing apps, but differ from popular apps, are more likely to succeed. We also highlight a novel asymmetry in how first-party entry by platform owners influences the performance of third-party apps, revealing that having first-party apps in the same space can either expand third-party output (App Store) or compete with it (Play Store). The remainder of this paper is organized as follows. In Section 2, we review the literature related to our analysis. Section 3 presents a dataset we use and descriptions of the clustering method. Section 4 shows the empirical analysis, and concluding remarks follow in Section 5.

2 Related Literature

This paper is closely related to the literature examining the determinants of success in the mobile application marketplace. A rich body of research has analyzed the characteristics of top-performing apps and identified factors that influence their likelihood of achieving high rankings in app stores. For instance, Lee and Raghu (2014) find that offering apps across multiple categories increases the probability of being featured in top charts, highlighting the importance of market scope and visibility. Similarly, Carare (2012) show that historical best-

seller status strongly affects current-period demand, also increasing consumers' willingness to pay. More recently, Picoto et al. (2019) report that while higher user ratings may negatively correlate with the likelihood of appearing among the top-50 grossing apps, other product-level characteristics—such as functionality and design complexity—positively affect success probabilities.

Another strand of the literature defines app success in terms of the number of downloads. Gokgoz et al. (2021) investigate the determinants of app success measured by downloads and find that appearing in top charts has the largest and most significant effect on demand, while app updates also exert positive effects—particularly for newly launched apps. Similarly, Sällberg et al. (2023) show that users' download decisions depend on rating and review information, including review length and sentiment. Our paper is closer to Bilal et al. (2024) which employ an unsupervised machine learning approach to cluster mobile apps based on their titles and find that apps with certain linguistic features in their names tend to achieve higher installation counts. Building on this idea, our paper clusters apps using textual data from their full descriptions and computes pairwise cosine similarity scores for every app. This method enables us to examine how an app's position in the horizontal product space contributes to its performance. We follow this stream of research by adopting app downloads as our measure of success and extend it by showing that app success also depends on where an app is positioned within the horizontal product space.[1] Using textual data from app descriptions, we demonstrate that apps' semantic proximity to others in the same market cluster systematically influences their performance outcomes.

This paper also relates to the growing literature on vertical integration and self-preferencing in two-sided platform markets. A longstanding debate concerns whether a platform's entry into downstream markets is pro-competitive or anti-competitive, with mixed evidence across contexts. Theoretical studies emphasize that the direction of the effect depends on the platform's strategic incentives. A platform may vertically integrate to exert pressure on downstream developers and thereby stimulate innovation or improve quality (Farrell and Katz, 2000; Gawer and Henderson, 2007), or engage in price squeezing that increases transactions and platform engagement (Hagiu et al., 2022). Conversely, the platform could be incentivized to improve its own product revenue, through foreclosing third-party sellers and distorting market competition (Rey and Tirole, 2007). Specifically, platforms may steer con-

[1]Previous research empirically documents a strong negative correlation between app rank and downloads. Garg and Telang (2013) model app downloads as a function of ranking under a Pareto distribution and find that top-ranked paid apps generate more than one hundred times the downloads of those ranked around 200, a finding replicated in subsequent work (Singh et al., 2024). Moreover, the likelihood of being listed among the top-50 grossing apps is largely determined by app downloads and in-app purchase revenues (Picoto et al., 2019). Accordingly, we treat monthly app downloads as a proxy for commercial success and as the key pathway toward achieving high-ranking status within the marketplace.

sumers toward their own products by raising commission fees (Anderson and Bedre-Defolie, 2024).

Empirical findings likewise suggest heterogeneous effects of platform entry across industries and categories. Foerderer et al. (2018) show that Google's launch of *Google Photos* app increased innovation among third-party photography apps, likely by attracting user attention that later translated into higher downloads; the spillover was especially pronounced for large and diversified developers.[2] Similarly, Li and Agarwal (2017) find that Facebook's integration of complementary services, which is *Instagram*, generated positive demand spillovers across related markets, though the effects benefited large apps more than smaller ones. In contrast, He et al. (2020) document that the entry of a Chinese e-commerce platform reduced offline demand for third-party retailers, with larger firms experiencing stronger declines. Consistent with foreclosure concerns, Zhu and Liu (2018) also show that Amazon's entry into product categories such as *Electronics* and *Home* displaced sales from third-party merchants, and Chen and Tsai (2024) recently demonstrate that Amazon's products are algorithmically recommended more often, a pattern not explained by consumer preferences but by the seller's identification.

While previous studies have shown that platform tends to target high-performing third-party products rather than weaker ones (e.g., Zhu and Liu (2018)), we examine how the *functional proximity* of first-party apps—that is, the horizontal space into which the platform integrates— affects the success of third-party apps. To the best of our knowledge, this is the first paper to analyze the impact of vertical integration through the lens of horizontal differentiation within the two-sided marketplace. This approach would also guide policymakers when regulating platform entry, such as in case where platform introduce a differentiated product from existing products that Bonazzi et al. (2025) find to be welfare-enhancing.

Finally, this paper contributes to the ongoing discussion on how to define market segments in mobile app stores. Most prior studies rely on platform-assigned categories to delineate competitive boundaries, while others cluster competing apps using Google Play's list of "similar apps" (Kesler et al., 2017; Ershov, 2018; Wen and Zhu, 2019). Our approach extends this line of research by drawing on the emerging literature that employs unsupervised machine learning techniques to infer market structure from textual data. In the industrial organization literature, for example, Hoberg and Phillips (2016) use natural language processing and cosine similarity to identify product-market competitors based on firms' business descriptions, whereas Leyden (2022) apply term frequency–inverse document frequency (TF–IDF) and X-means clustering to define market boundaries. Building on these methodologies, our paper measures pairwise cosine similarity using TF–IDF representations and applies a hi-

[2] "Diversified" refers to developers offering apps in multiple categories.

erarchical clustering algorithm following Hoberg and Phillips (2016) to define economically meaningful market segments among mobile apps.

3 Data

In this section, we describe the dataset we use and how we cluster apps based on their feature descriptions. We utilize a monthly app-level panel dataset for the global market, spanning from January 2012 to May 2021, which was collected from Sensor Tower, an AI-driven mobile analytics firm. The dataset includes core performance measures such as the number of app downloads, updates, and revenues, along with app-specific attributes including user ratings, the number of ratings, app price, textual descriptions, and release dates.

During the data pre-processing stage, we exclude a substantial portion of observations. Specifically, we drop apps with missing information on key variables as well as those whose descriptions are written in non-English languages, eventually excluding fringe apps. We consider this restriction to be justified, given that this paper primarily focuses on the determinants of apps' success and the market has a right-skewed distribution of app downloads. The resulting dataset is a panel dataset comprising 15,743 unique apps from the App Store (September 2014–May 2021) and 29,546 unique apps from the Play Store (August 2014–May 2021). Table 1 presents the summary statistics for the main variables in each app store and Table B.3 and Table B.4 show the correlations between the variables we use. We also report lists of first-party apps in each store in the Appendix B.

3.1 Defining markets

Platform-assigned categories are coarse and often blur competitive boundaries; for example, music streaming and karaoke apps are grouped under "Music" despite weak substitutability. On the other hand, some of the entertainment apps and gaming apps would be substitutable to some degree, while they are assigned to different categories. To recover more meaningful market definitions, we construct a text-based similarity measure from app descriptions using pairwise cosine similarity. We then cluster apps with high similarity scores to create new competitive segments. This procedure parallels text-based industry classification approaches in Hoberg and Phillips (2016), where firm descriptions are mapped into a continuous product space to refine market boundaries.

We begin by applying Natural Language Processing (NLP) to convert unstructured app descriptions into numerical representations. We first construct a vocabulary of all unique terms and represent each app by a TF–IDF vector, where term frequency (TF) captures the

frequency of a word within the description and inverse document frequency (IDF) adjusts for its uniqueness across all apps. This weighting scheme reduces the influence of ubiquitous words while emphasizing distinctive terms, thereby sharpening the measurement of semantic proximity. Pairwise cosine similarity is then computed from the TF–IDF vectors, providing a quantitative representation of the semantic similarity between apps. This approach allows us to refine broad platform-assigned categories into finer market partitions that better capture actual competitive relationships among apps.[3]

3.1.1 Clustering with cosine similarity

Following the NLP preprocessing, the document–term matrix remains high-dimensional and sparse, rendering direct similarity calculations computationally costly. To address this, we apply truncated singular value decomposition (SVD) to reduce dimensionality, and then normalize each document vector to unit length. Pairwise cosine similarities are subsequently computed from the transformed TF–IDF vectors.

Formally, let $\mathbf{tf\text{-}idf}_j = (\text{tf-idf}_{j1}, \ldots, \text{tf-idf}_{jT})$ denote the TF-IDF vector of app j for word terms $t = 1, 2, \cdots, T$. We normalize this vector using L2 normalization:

$$S_j = \frac{\mathbf{tf\text{-}idf}_j}{\|\mathbf{tf\text{-}idf}_j\|_2} = \frac{\mathbf{tf\text{-}idf}_j}{\sqrt{\sum_{t=1}^{T}(\text{tf-idf}_{jt})^2}}. \tag{1}$$

Here, S_j denotes the normalized representation of app j. The cosine similarity between apps j and k is then given by the dot product of their normalized vectors:

$$SIM_{jk} = (S_j \cdot S_k). \tag{2}$$

Using the cosine similarity vectors, we define market segments through a bottom-up agglomerative hierarchical clustering procedure, analogous to the text-based industry classification in Hoberg and Phillips (2016). The main steps of the clustering algorithm are outlined below:

1. With sub-samples of apps, define initial markets by assigning each app to its own singleton cluster.

2. Merge the two clusters with the highest cosine similarity, forming a new cluster with combined membership.

[3]TF–IDF has been widely applied in information retrieval and text summarization to assess word importance (e.g., Bun and Ishizuka, 2002; Nenkova and McKeown, 2012; Christian et al., 2016). Appendix A provides further details.

3. Recompute similarities between all clusters. For clusters with multiple members, the average cosine similarity is defined as

$$I_{lg} = \sum_{x=1}^{m_l} \sum_{y=1}^{m_g} \frac{SIM_{xy}}{m_l m_g},$$

where m_l and m_g denote the number of apps in clusters l and g, respectively. And SIM_{xy} is the cosine similarity between apps $x \in l$ and $y \in g$.

4. Merge the pair of clusters with the highest average similarity.

5. Repeat steps 3–4 until the resulting within-cluster similarity falls below either 1) the similarity level of the previously merged clusters or 2) a predetermined threshold ρ.

The clustering algorithm requires specifying both the sub-samples and the similarity threshold (ρ). We define sub-samples using the distribution of cumulative downloads as of May 2021, the final period in our dataset. As shown in Figure 1, an elbow appears around the 90th percentile in both the App Store and the Play Store. Accordingly, we restrict the initial sub-sample to apps below this cutoff. This restriction is justified by the highly right-skewed distribution of downloads, where a small fraction of apps captures the vast majority of the market. We then set the within-cluster similarity threshold to $\rho = 0.15$ for the App Store and $\rho = 0.10$ for the Play Store, which corresponds to approximately the 85th - 90th percentile of the pairwise cosine similarity distribution (Table 2), thereby retaining only the most similar quintile of app pairs. Applying this procedure yields 108 clusters for the App Store and 107 clusters for the Play Store.

4 Empirical Anlayis

4.1 Estimation

The primary objective of this study is to investigate the key factors that contribute to the success of third-party apps. We hypothesize that an app's position within the horizontal and vertical product spaces, as well as the presence of first-party apps, jointly contribute to its market outcomes. Horizontal differentiation is quantified by the average cosine similarity between app j and all other apps within its cluster, denoted as SIM_{jct}. A smaller SIM_{jct} indicates that app j is more distinct relative to its cluster peers. In addition, we compute TOP_SIM_{jct}, the average cosine similarity between app j and the top-performing

apps within the clusters.[4] Both measures evolve over time as new entries alter the similarity structure within clusters. Vertical differentiation is proxied by the number of app updates, $updates_{jct}$, which captures the developer's investment in quality improvements and maintenance. Finally, to capture the potential influence of first-party app, we include measures of first-party activity, defined by (i) $First_{ct}$, an indicator for the presence of a first-party app in cluster c at month t, and (ii) the average age of such apps $First\_Avg\_Age_{ct}$.[5]

Our baseline empirical specification is given by:

$$y_{jct} = \alpha_1 SIM_{jct} + \alpha_2 TOP\_SIM_{jct} + \alpha_3 Updates_{jct} + \alpha_4 First_{ct} + \boldsymbol{X}_{jct}\boldsymbol{\beta} + \delta_j + \delta_c + \delta_t + \varepsilon_{jct}, \tag{3}$$

where y_{jct} denotes the outcome variable—the natural logarithm of downloads—for app j in cluster c at time t. The vector $\boldsymbol{X}_{jct}$ represents a set of app-month-level controls, including app age, cumulative user ratings, price, and average star rating.[6] App fixed effects (δ_j) absorb time-invariant heterogeneity across applications, cluster fixed effects (δ_c) capture market-segment-level characteristics, and year-month fixed effects (δ_t) control for aggregate monthly shocks. The error term ε_{jct} represents unobserved, idiosyncratic demand shocks.

4.2 Identification

Our coefficients of interest are $\vec{\alpha} = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$. A primary identification concern is that unobserved shocks may simultaneously influence both the entry of new applications and the demand for existing ones. For instance, a sudden surge in public attention to cryptocurrencies could lead to a contemporaneous rise in downloads of investment or banking apps while also prompting new entrants to launch similar products within the same market segment. This simultaneity implies that observed variations in within-cluster similarity could partially capture correlated demand shocks rather than true changes in horizontal differentiation. However, since the design, coding, and approval of a new app typically require several months, short-lived demand shocks are unlikely to immediately trigger new entries, attenuating the potential bias from transitory shocks.[7]

[4] These top apps correspond to the seed apps used in defining the clusters.

[5] The indicator varies over time if a first-party app is introduced during the observation period.

[6] To capture potential network effects, we employ the logarithm of the cumulative number of user ratings from the previous month. A positive coefficient would be consistent with the presence of network externalities, reflecting economies of scale, social influence, or learning effects (Peng et al., 2011). Comparable identification strategies in the literature exploit exogenous shocks to user networks, such as the introduction of Game Center on the App Store (Boudreau et al., 2022) or changes in mobile penetration and churn rates (Doganoglu and Grzybowski, 2007).

[7] Typically, the mobile app development process takes 3-12 months (e.g., Ghandi et al. (2017)). Even for an app equipped with back-end infrastructure, Singh et al. (2024) assumes it takes a month to develop front-

Moreover, our empirical design incorporates a rich set of fixed effects at the app, cluster, and year–month levels, which absorb persistent heterogeneity and common market shocks. While developers may respond to persistent demand trends, we assume that such anticipatory effects are unlikely to fully align with month-to-month shocks. While short-run changes in similarity mechanically arise from new app entries, the assumption implies that these entries are not immediate responses to transitory demand shocks. Instead, they reflect developers' medium-term strategic or investment decisions, which are unlikely to be synchronized with month-to-month fluctuations in app demand. We further assess the robustness of these assumptions through additional tests reported in Section 4.5.

Unlike the third-party app entry, the relationship between the first-party entry and unobserved demand shocks in y_{jct} is inherently ambiguous. Platforms may hold private, app-specific information—such as granular user engagement data—that is inaccessible to external developers. If a particular app experiences a surge in popularity, the platform could exploit this proprietary information and strategically launch its own competing app, creating a correlation between first-party entry and latent demand shocks. Furthermore, reverse causality may arise for the app updates variable because apps with larger user bases and download volumes tend to release updates more frequently, further complicating identification.

To mitigate these endogeneity concerns, we construct a set of instrumental variables based on lagged cluster-level averages of competitors' characteristics:

1. the $t-2$ lag of the average app ratings, $\sum_{k \in \mathcal{C} \setminus \{j\}} rating_{kct-2}$,
2. the $t-2$ lag of the average revenues of competitors, $\sum_{k \in \mathcal{C} \setminus \{j\}} revenue_{kct-2}$,
3. the $t-1$ lag of the average number of updates, $\sum_{k \in \mathcal{C} \setminus \{j\}} Updates_{kct-1}$,
4. the $t-1$ lag of market concentration, proxied by the Herfindahl–Hirschman Index (HHI), computed as $\sum_{k \in \mathcal{C} \setminus \{j\}} share^2_{kct-1}$,
5. and, finally, the developer's own lagged update activity on its other apps during the previous month.

Instrument validity The use of competitors' past characteristics is in the same spirit as BLP instruments (Berry et al., 1995) that mitigates simultaneity concerns by ensuring that the instruments are predetermined with respect to current download shocks. The validity of our instruments rests on two standard conditions. Under the exclusion restriction, the

end in practice. This is because as Raj (2021) notes, release decisions are planned in advance and depend on time-consuming creative and development processes, reducing the likelihood that short-lived demand fluctuations immediately translate into new app launches.

lagged instruments capture rivals' historical performance and strategic investments but are unlikely to be correlated with app j's unforeseen demand shocks once app, cluster, and time fixed effects are absorbed, except through their influence on updating or first-party entry.[8] For example, *Spotify*'s past ratings or revenues would not be directly affected by contemporaneous demand shocks for *YouTube Music*. Likewise, a developer's past update intensity across its portfolio provides a supply-side shifter that is orthogonal to short-run demand shocks for a specific app j.

The instruments are also expected to satisfy the relevance condition. Lagged ratings and revenues of competitors convey profitability signals and market opportunities that guide both developers' updating behavior and the platform's first-party entry decisions. Lagged update intensity and market concentration, which we measure with the Herfindahl–Hirschman Index (HHI), capture adjustment costs and strategic timing, while the developer's own lagged update activity reflects persistent managerial capacity and release frequency. These features make the instruments strong predictors of current updating and platform entry activity. Empirically, the F-statistics in the first stage exceed the conventional threshold of 10 across all specifications, indicating that weak instrument concerns are minimal. Overall, the proposed instruments are theoretically justified and empirically supported, satisfying the identifying assumptions required for consistent estimation of $\vec{\alpha}$.

A remaining concern is that first-party app entry could be computationally correlated with within-cluster cosine similarity, potentially confounding the estimated effect of platform entry on app performance. In particular, the introduction of a first-party app may simultaneously alter the similarity structure within a cluster and directly affect the dependent variable. If such a correlation were substantial, it could inflate the estimated effect of first-party entry or attenuate the role of differentiation measures. Empirically, however, this concern appears limited. As shown in Tables B.3 and B.4, the pairwise correlation between first-party entry and cosine similarity is low, and the variance inflation factors (VIFs) across all specifications remain below 3, suggesting that multicollinearity is not a serious issue.

Furthermore, we argue that first-party apps should be incorporated in the computation of cosine similarity. Considering that the dependent variable reflects the realized competitive environment—including the presence of first-party apps—excluding them from the similarity computation would create a mismatch between the explanatory and outcome variables, leading to omitted-variable bias. From an economic standpoint, first-party apps often represent prominent competitors within their clusters; omitting them would artificially reduce the measured degree of competition and understate the true level of horizontal overlap captured

[8] All cluster-level instruments are computed in a *leave-one-out* fashion, ensuring that app j's own outcomes do not mechanically enter the instrument set.

by the similarity index.[9] Finally, we verify the robustness of this inclusion decision by recalculating cosine similarity after excluding all first-party apps (Table 5); the results remain qualitatively unchanged, reinforcing that our baseline findings are not driven by mechanical collinearity between first-party entry and similarity measures.

4.3 Results

In this section, we present the estimation results from equation (3) separately for the two app store datasets. Table 3 reports the results for the App Store sample, and Table 4 shows the corresponding estimates for the Play Store. While both analyses address the same identification challenges discussed earlier, the specific sets of instrumental variables differ slightly across platforms. For the App Store, we instrument the endogenous regressors using lagged averages of competitors' app ratings, the Herfindahl–Hirschman Index (HHI) of competitors' download shares, and the previous month's updates of the same developer's other apps. For the Play Store, we instead use lagged averages of competitors' ratings and updates, as well as past revenues of top-performing apps within each cluster.

We report four specifications for each dataset. Columns (1) and (2) present the OLS estimates, while columns (3) and (4) show the corresponding two-stage least squares (2SLS) results using our instrumental-variable approach. A key difference emerges in the estimated coefficients of $First\_Party_{ct}$. Under the 2SLS specification, the coefficients become more economically significant for both platforms, but the signs switch from positive to negative for the Play Store while remaining positive for the App Store. These patterns suggest that Apple's (Google's) first-party app entry is subject to downward (upward) bias under OLS estimation, implying that there would be a negative (positive) correlation between the platform's entry decision and unobserved third-party app-specific demand shocks. In other words, failing to account for endogeneity would understate Apple's positive competitive effects and fail to recognize Google's negative effects.

The empirical diagnostics further validate our identification strategy. The first-stage regressions yield large F-statistics, indicating that weak instrument concerns are minimal. The Wu–Hausman tests reject the null hypothesis of exogeneity at the 1% level, confirming that the regressors of interest are indeed endogenous and that 2SLS estimation is warranted. Moreover, the Sargan over-identification tests do not reject the null, consistent with the exclusion restriction. Taken together, these results provide strong evidence that our instruments are both relevant and valid. Although the qualitative patterns across columns (3) and (4) are similar, we treat column (3) as our preferred specification, as it exhibits stronger

[9] As reported in Tables B.1 and B.2, 4 of Apple's 10 first-party apps and 29 of Google's 42 first-party apps are classified as leading apps within their respective clusters.

instrument validity and robustness across specifications.

Results in Tables 3 and 4 show that for both the App Store and the Play Store, apps with higher average cosine similarity to other apps within the same cluster—that is, apps that are less horizontally differentiated—tend to attract more downloads. In contrast, a higher similarity to the cluster's top-performing apps is associated with fewer downloads, suggesting that proximity to the most successful competitors results in customers choosing the rival, thus reducing app success. Consistent with the view that product improvement enhances demand, the number of updates exhibits a positive and significant effect on app downloads on both platforms. The presence of Apple's own app in a cluster also has a positive effect on third-party app performance, and the average age of Apple's first-party apps further amplifies this effect. These results could imply that Apple's entry may generate positive spillovers—possibly through demand expansion or market validation—for third-party developers operating in the same product space. We discuss this further in Section 4.4. In contrast, the presence of Google's own app is associated with a decline in downloads for third-party apps within the same cluster, suggesting that Google's entry exerts a crowding-out effect or a stronger competitive effect that offsets positive externalities. We also examine this further in Section 4.4.

The results collectively indicate that apps positioned to be similar to the average app in a market segment, but still some distance from the most popular app, tend to achieve higher download volumes. This might be understood to represent a moderate differentiation strategy: close, but not too close. It might also be that apps cluster where customer demand is greatest, so that being closer to the average app means the app under consideration is locating near where customer surplus is greatest. But at the same time, being too much like the most popular app means that the app in question lacks a sufficiently unique value proposition. To illustrate, consider a developer preparing to launch a new health app within a *Fitness & Health* cluster. To maximize success, the developer should preserve the core features that define the cluster's identity—ensuring the app remains recognizable to customers that value such health apps—while introducing novel attributes that distinguish it from the market leader. Such positioning allows the app to benefit from shared consumer demand within the cluster without being overshadowed by leading competitors.

4.4 First-party apps

The coefficients on $First\_Party_{ct}$ in equation (3) capture the marginal effect of the presence of first-party apps on third-party app performance. In this section, we examine whether this impact varies with a third-party app's horizontal proximity to the first-party app within the

cluster. This analysis allows us to examine possible effects of how customers identify and choose apps, and of platform provider strategies. It might be that the presence of first-party apps stimulates sales of third-party apps. Reasons for this might include: (1) The presence stimulates customer interest in such apps; (2) Rivals interpret the presence as a signal that the first-party has proprietary knowledge about customer demand; and (3) The presence is part of the first-party's strategy to profit from platform value, which is enhanced by the value customers place on the apps present. It might also be that the presence of first-party apps depresses third-party app sales, perhaps because the first-party app is a substitute for some third-party apps, or because the platform sabotages rivals.

Formally, we estimate the following specification:

$$\begin{aligned} y_{jct} = {} & \gamma_1 SIM\_{}_{jct}^{EX} + \gamma_2 TOP\_SIM\_{}_{jct}^{EX} + \gamma_3 Updates_{jct} + \gamma_4 SIM\_First_{ct} \\ & + \boldsymbol{X}_{jct}\boldsymbol{\omega} + \delta_j + \delta_c + \delta_t + \varepsilon_{jct}, \end{aligned} \tag{4}$$

where SIM_First_{ct} denotes the within-cluster cosine similarity between app j and the first-party app(s) in cluster c at time t. This measure isolates not only the presence of a first-party app but also how its entry affects neighboring third-party apps that are closer in horizontal space. Consequently, $SIM\_{}_{jct}^{EX}$ and $TOP\_SIM\_{}_{jct}^{EX}$ are computed without the first-party apps. To further disentangle the role of horizontal positioning itself, we construct a counterfactual measure, $SIM\_First\_Like_{ct}$, by assigning each cluster a hypothetical first-party app. Specifically, in each cluster, we identify one of the leading apps that exhibits the highest pairwise cosine similarity with the actual first-party app and designate it as the "first-like" app. We then replace SIM_First_{ct} in equation (4) with the corresponding cosine similarity between app j and this hypothetical app. The variable, $SIM\_First\_Like_{ct}$, equals zero when no first-party app is present and takes positive values otherwise, thereby allowing us to separate the effects of entry per se from those arising through spatial proximity to the first-party app.

Table 5 reports the estimation results using the same identification strategy described in Section 4.2. Coefficients for key parameters remain unchanged in both magnitude and significance, further reinforcing our main findings. Notably, the coefficient on SIM_First_{ct} is statistically significant and larger in magnitude than that of $First\_Party_{ct}$ reported in Tables 3 and 4. This suggests that not only the presence of a first-party app, but also its relative location in the horizontal product space, plays an important role in shaping third-party app performance. In the case of Apple (Google), closer proximity leads to greater (diminished) third-party app success. The results for $SIM\_First\_Like_{ct}$ provide additional support for this interpretation. Even when we replace the actual first-party app with a hypothetical

third-party app that most closely resembles it, the estimated effects remain consistent in sign and magnitude, underscoring the robustness of the proximity-based mechanism.

To further illustrate these findings, Figure 2 compares the distributions of within-cluster average cosine similarities between clusters with and without first-party apps. The figure reveals distinct spatial patterns across platforms. Apple's first-party apps tend to be located in clusters with higher average similarity—indicating less horizontal differentiation. In other words, when Apple provides an app, it places the app in spaces that rivals also find valuable. In contrast, Google's apps are typically positioned in more differentiated clusters. Moreover, the average similarity between third-party apps and Apple's own apps is higher than that between third-party apps and Google's, suggesting that Apple's products are more closely aligned with third-party apps' features.

These patterns imply that Apple's first-party apps are positioned in market segments that overlap substantially with third-party apps, and that this positioning is associated with greater third-party success. This might be that Apple apps expand consumer awareness and demand within those clusters (Li and Agarwal, 2017; Foerderer et al., 2018). It might also be that people who choose iPhones are different from and more homogeneous than those who choose Android phones.

Further supporting the finding that buyers of iPhones are different from buyers of Android phones, Table 5's results imply that, in general, customers are more responsive to product differences on Google's platform than on Apple's. Coefficients for customer responses to updates, for example, are over six times larger in the Play Store than in the App Store. Also, the effects of similarity to the most popular apps are different. While the coefficients are negative and statistically significant on both platforms—likely indicating substitution effects—the effects on Google's platform are four to eight times those on Apple's. This is not conclusive proof that the customers are substantially different—it might be that features on Android phones make customers more responsive—but it is consistent with the conclusions of industry observers: iPhone users spend more on apps, are more likely to use productivity and social media apps, and less likely to use education and food and drink apps than their Android counterparts (Szczygieł, B, 2025). Additionally, iPhone users tend to have higher incomes, higher education levels, and are more likely to be men than Android users (Buildfire, 2024).

In contrast with Apple, Google's first-party apps are located in relatively distinct product spaces, with novel features that may steer market demand and reallocate user attention rather than broadening the market. Consistent with prior studies, the entry of Google's apps could exert a competitive squeeze (Farrell and Katz, 2000), prompting developers to re-optimize their innovation and investment strategies (Belleflamme and Peitz, 2019). From

a welfare perspective, when a first-party app is more differentiated than third-party apps, platform entry could be welfare-enhancing (Bonazzi et al., 2025).

Differences in app performance on Apple versus Google could also reflect differences in the two firms' business models. Google is a software company and receives nearly 80 percent of its revenue from advertising (Statista, 2025). Apple has been largely a computer hardware company. Although its services revenue is growing in importance, it is still only 21 percent of the company's revenue (Levy, A, and Leswing, K, 2024).

4.5 Robustness Check

We conduct several robustness checks to assess the exogeneity of the cosine similarity measures. In particular, we re-estimate equation (3) using lagged values of within-cluster similarity variables, thereby mitigating potential simultaneity between app downloads and contemporaneous changes in similarity. Tables 6 and 7 present the results for the App Store and Play Store, respectively. The estimated coefficients remain stable in magnitude and significance relative to our baseline results, indicating that our main findings are not sensitive to alternative timing assumptions.[10] Furthermore, Table 8 reports the distribution of within-app autocorrelations for the similarity measures. Both SIM and TOP_SIM exhibit strong temporal persistence, with median first-order autocorrelations (ρ_1) exceeding 0.7 and second-order autocorrelations (ρ_2) above 0.5 for both platforms. This high degree of persistence suggests that the similarity indices evolve gradually over time and are unlikely to be driven by short-lived or transitory shocks. Together, these findings support our interpretation of cosine similarity as an exogenous, structural characteristic of app positioning within clusters, rather than an outcome that is jointly determined with app downloads.

5 Discussion

This study proposes a novel framework for analyzing success in mobile application markets through the lens of horizontal differentiation in app functionalities. By applying natural language processing to app descriptions, we construct pairwise cosine similarity measures across third-party apps and use these measures to cluster apps based on functional proximity.

[10] In the main specification, the Sargan over-identification tests fail to reject the null hypothesis, confirming the validity of the instruments. In several robustness checks, however, the Sargan statistic becomes significant. This pattern likely reflects the reduced precision and sensitivity of the exclusion restrictions under those specifications rather than a fundamental violation of instrument exogeneity. Importantly, the estimated coefficients remain qualitatively consistent across all models, reinforcing that our conclusions are not driven by instrument validity concerns.

These data-driven market boundaries allow us to identify key determinants of app demand that are not observable under platform-defined categories.

Our results show that developers are more likely to succeed when entering product spaces in which their apps share core functionalities with competing apps, to attain high valuations from the cluster users, while remaining differentiated from the most popular rivals. We further find a pronounced asymmetry in the effects of first-party entry between platforms. Specifically, the presence of first-party apps can either enhance third-party success, as observed in the App Store, or diminish it, as in the Play Store, with the results also depending on the entrant's horizontal positioning strategy. These findings offer actionable insights for developers regarding market selection and optimal differentiation strategies.

Our findings also carry implications for platform regulation and competition policy in digital markets. First, regulations in the EU and those proposed in the US aim to make Apple's platform policies more similar to those of Google. But Apple's overall platform strategy appears to be beneficial to third-party apps in our analysis. Second, the asymmetric effects of first-party entry across platforms suggest that blanket regulatory assessments of vertical integration may defeat their purposes. Policies that seek to control platform conduct solely based on the presence of first-party apps or commission structures risk overlooking how the entrant's horizontal positioning shapes competitive outcomes. Third, our results highlight the importance of incorporating product differentiation into antitrust and regulatory analyses, rather than focusing exclusively on prices or entry decisions. In some situations, first-party entry that closely overlaps with existing third-party apps may expand third-party sales, whereas entry into highly differentiated spaces may redirect demand and diminish third-party sales. Accordingly, *ex ante* regulation should account for differences between platforms and app differentiation strategies

Several promising avenues for future research remain. First, incorporating dynamic horizontal differentiation would allow for a richer analysis of how app features evolve over time. As emphasized by Hoberg and Phillips (2016), firms frequently update product characteristics, and mobile applications may similarly adjust their functionalities—changes that are not directly observed in our data. Second, examining app exit decisions in response to intensified horizontal competition or first-party entry would further enhance our understanding of market dynamics and competitive pressure in mobile app ecosystems.

References

Simon Anderson and Ozlem Bedre-Defolie. Hybrid platform model: Monopolistic competition and a dominant firm. RAND Journal of Economics, 2024.

Apple. Your apps on the app store, 2025a. URL https://developer.apple.com/app-store-connect/. Accessed: 2025-12-23.

Apple. App review guidelines, 2025b. URL https://developer.apple.com/app-store/review/guidelines/. Accessed: 2025-12-23.

Apple. Membership details, 2025c. URL https://developer.apple.com/programs/whats-included/. Accessed: 2025-12-23.

Apple. Xcode, 2025d. URL https://developer.apple.com/xcode/. Accessed: 2025-12-23.

Paul Belleflamme and Martin Peitz. Managing competition on a two-sided platform. Journal of Economics & Management Strategy, 28(1):5–22, 2019.

Steven Berry, James Levinsohn, and Ariel Pakes. Automobile prices in market equilibrium. Econometrica, 63(4):841–890, 1995. URL https://www.jstor.org/stable/2171802.

Ahmad Bilal, Hamid Turab Mirza, Ibrar Hussain, and Adnan Ahmad. Investigating influence of google-play application titles on success. Big Data Research, 36:100443, 2024.

Leda Maria Bonazzi, Riccardo Martina, and Giovanni Ursino. Hybrid platforms with free entry: demand-enhancing activities. Journal of Industrial and Business Economics, 52(1): 81–106, 2025.

Kevin J Boudreau, Lars Bo Jeppesen, and Milan Miric. Competing on freemium: Digital competition with network effects. Strategic Management Journal, 43(7):1374–1401, 2022.

Buildfire. iphone users vs. android users: How do they behave differently?, 2024. URL https://buildfire.com/ios-android-users/. Accessed: 2025-12-30.

Khoo Khyou Bun and Mitsuru Ishizuka. Topic extraction from news archive using tf* pdf algorithm. In Proceedings of the Third International Conference on Web Information Systems Engineering, 2002. WISE 2002., pages 73–82. IEEE, 2002.

Robert D Buzzell. Are there "natural" market structures? Journal of Marketing, 45(1): 42–51, 1981.

Octavian Carare. The impact of bestseller rank on demand: Evidence from the app market. International Economic Review, 53(3):717–742, 2012.

Nan Chen and Hsin-Tien Tsai. Steering via algorithmic recommendations. The RAND Journal of Economics, 55(4):501–518, 2024.

Hans Christian, Mikhael Pramodana Agus, and Derwin Suhartono. Single document automatic text summarization using term frequency-inverse document frequency (tf-idf). ComTech: Computer, Mathematics and Engineering Applications, 7(4):285–294, 2016.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. Journal of the American society for information science, 41(6):391–407, 1990.

Toker Doganoglu and Lukasz Grzybowski. Estimating network effects in mobile telephony in germany. Information Economics and Policy, 19(1):65–79, 2007.

Daniel Ershov. Competing with superstars in the mobile app market. Available at SSRN 3265662, 2018.

Joseph Farrell and Michael L Katz. Innovation, rent extraction, and integration in systems markets. The journal of industrial economics, 48(4):413–432, 2000.

Jens Foerderer, Thomas Kude, Stefan W. Schuetz, and Armin Heinzl. Does platform owner's entry crowd out innovation? evidence from google play. Information Systems Research, 29(2):444–460, 2018. doi: 10.1287/isre.2018.0787.

Rajiv Garg and Rahul Telang. Inferring app demand from publicly available data. MIS quarterly, pages 1253–1264, 2013.

Annabelle Gawer and Rebecca Henderson. Platform owner entry and innovation in complementary markets: Evidence from intel. Journal of Economics & Management Strategy, 16(1):1–34, 2007.

Lizeth Ghandi, Catarina Silva, Danilo Martínez, and Tatiana Gualotuña. Mobile application development process: A practical experience. In 2017 12th Iberian Conference on Information Systems and Technologies (CISTI), pages 1–6. IEEE, 2017.

Zeynep Aydin Gokgoz, M Berk Ataman, and Gerrit H van Bruggen. There's an app for that! understanding the drivers of mobile application downloads. Journal of Business Research, 123:423–437, 2021.

Andrei Hagiu, Tat-How Teh, and Julian Wright. Should platforms be allowed to sell on their own marketplaces? The RAND Journal of Economics, 53(2):297–327, 2022.

Shu He, Jing Peng, Jianbin Li, and Liping Xu. Impact of platform owner's entry on third-party stores. Information Systems Research, 31(4):1467–1484, 2020.

Gerard Hoberg and Gordon Phillips. Text-based network industries and endogenous product differentiation. Journal of political economy, 124(5):1423–1465, 2016.

Baojun Jiang, Kinshuk Jerath, and Kannan Srinivasan. Firm strategies in the "mid tail" of platform-based retailing. Marketing Science, 30(5):757–775, 2011.

Reinhold Kesler, Michael Kummer, and Patrick Schulte. User data, market power and innovation in online markets: Evidence from the mobile app industry. Industry and Innovation Journal, 2, 2017.

Gunwoong Lee and T Santanam Raghu. Determinants of mobile apps' success: Evidence from the app store market. Journal of Management Information Systems, 31(2):133–170, 2014.

Levy, A, and Leswing, K. Apple's gross margin hits record as services business keeps growing. Technical report, CNBC, 2024. URL `https://www.cnbc.com/2025/01/30/apples-gross-margin-hits-record-as-services-business-keeps-growing.html`.

Benjamin T Leyden. There's an app (update) for that. Technical report, Technical report, Working Paper, 2022.

Xinxin Li and Anuj Agarwal. Platform integration and demand spillovers in complementary markets: Evidence from facebook. Management Science, 63(10):3438–3458, 2017.

Christopher D Manning. Introduction to information retrieval. Syngress Publishing,, 2008.

Ani Nenkova and Kathleen McKeown. A survey of text summarization techniques. In Mining text data, pages 43–76. Springer, 2012.

Khumaisa Nur'Aini, Ibtisami Najahaty, Lina Hidayati, Hendri Murfi, and Siti Nurrohmah. Combination of singular value decomposition and k-means clustering methods for topic detection on twitter. In 2015 International conference on advanced computer science and information systems (ICACSIS), pages 123–128. IEEE, 2015.

Gang Peng, Ming Fan, and Debabrata Dey. Impact of network effects and diffusion channels on home computer adoption. Decision Support Systems, 51(3):384–393, 2011.

Winnie Ng Picoto, Ricardo Duarte, and Inês Pinto. Uncovering top-ranking factors for mobile apps through a multimethod approach. Journal of Business Research, 101:668–674, 2019.

Manav Raj. Friends in high places: Demand spillovers and competition on digital platforms. Available at SSRN, 3843249, 2021.

Patrick Rey and Jean Tirole. A primer on foreclosure. In Handbook of Industrial Organization, Vol. 3, pages 2145–2220. Elsevier, 2007.

Henrik Sällberg, Shujun Wang, and Emil Numminen. The combinatory role of online ratings and reviews in mobile app downloads: an empirical investigation of gaming and productivity apps from their initial app store launch. Journal of Marketing Analytics, 11(3): 426–442, 2023.

Amandeep Singh, Kartik Hosanagar, and Aviv Nevo. Network externalities and app store fees in mobile platforms. Available at SSRN 3911638, 2024.

Szczygieł, B. ios vs android: Key differences in 2025, 2025. URL `https://www.netguru.com/blog/iphone-vs-android-users-differences`. Accessed: 2025-12-30.

Wen Wen and Feng Zhu. Threat of platform-owner entry and complementor responses: Evidence from the mobile app market. Strategic Management Journal, 40(9):1336–1367, 2019.

Christopher S Yoo, Jesse Lambert, and Timothy P Pfenninger. Municipal fiber in the united states: A financial assessment. Telecommunications Policy, 46(5):102292, 2022.

Nan Zhong and Florian Michahelles. Google play is not a long tail market: An empirical analysis of app adoption on the google play app market. In Proceedings of the 28th annual ACM symposium on applied computing, pages 499–504, 2013.

Feng Zhu and Qihong Liu. Competing with complementors: An empirical look at amazon.com. Strategic Management Journal, 39(10):2618–2642, 2018.

6 Tables

Table 1: Summary Statistics by Platform

	App Store		Play Store	
Variable	Mean	Std	Mean	Std
Panel A. App-level variables				
App downloads	28,198	371,428	62,360	748,815
Number of app updates	0.423	0.944	0.437	0.929
Multihoming	0.465	0.499	0.368	0.482
App price ($)	0.690	7.445	0.180	1.845
Number of app ratings	9,300	132,479	45,579	964,744
Average app rating	3.750	0.840	4.03	0.55
App age (months)	55.192	31.874	38.00	27.26
Panel B. Platform-level variables				
Presence	0.058	0.233	0.242	0.428
Total number of app	0.058	0.233	0.446	1.035
Average app age (months)	3.165	14.800	12.029	24.399

Notes: Table reports summary statistics for apps available on Apple's App Store and Google Play Store. App-level statistics are based on 546,269 (App Store) and 712,548 (Play Store) observations. Monetary values are in U.S. dollars. The platform-level variables are based on the cluster-month level. For example, Google's first-party apps are present in 24% of clusters across the time periods, on average. Table B.2 shows in which clusters Google has entered.

Table 2: Distributions of Pairwise Cosine Similarities

	75%	90%	95%
App Store	0.0806	0.1539	0.2220
Play Store	0.0606	0.1267	0.1976

Table 3: Determinants of third-party apps performance in App Store

Dependent Variable:	$log(downloads_{jct})$			
Model:	(1) OLS	(2) OLS	(3) 2SLS	(4) 2SLS
Variables				
SIM_{jct}	0.6373*	0.6909*	0.6569*	0.8062**
	(0.3679)	(0.3688)	(0.3480)	(0.3498)
TOP_SIM_{jct}	-0.8522***	-0.7991***	-0.8991***	-0.7117***
	(0.2440)	(0.2430)	(0.2416)	(0.2360)
$Updates_{jct}$	0.1260***	0.1253***	0.5764***	0.4994***
	(0.0040)	(0.0040)	(0.1242)	(0.1299)
$First\_Party_{ct}$	0.2224**		1.296***	
	(0.0909)		(0.3758)	
$First\_Avg\_Age_{ct}$		0.0064***		0.0154***
		(0.0011)		(0.0046)
App_price_{jct-1}	-0.0456***	-0.0455***	-0.0444***	-0.0444***
	(0.0155)	(0.0156)	(0.0151)	(0.0152)
$Cum\_Ratings_{jct-1}$	0.4337***	0.4326***	0.4223***	0.4251***
	(0.0100)	(0.0100)	(0.0096)	(0.0094)
App_Rating_{jct}	-0.0950***	-0.1035***	-0.1631***	-0.1602***
	(0.0231)	(0.0227)	(0.0250)	(0.0247)
App_Age_{jct}	-0.0138***	-0.0131***	-0.0122***	-0.0107***
	(0.0034)	(0.0036)	(0.0030)	(0.0032)
Fixed-effects				
Cluster ID	Yes	Yes	Yes	Yes
Month	Yes	Yes	Yes	Yes
App	Yes	Yes	Yes	Yes
Fit statistics				
Observations	546,269	546,269	546,269	546,269
R^2	0.84925	0.84943	0.82991	0.83627
F-statistics ($Updates_{jct}$)			356.3***	356.3***
F-statistics ($First\_Party_{ct}$)			12,079.7***	
F-statistics ($First\_Avg\_Age_{ct}$)				13,337.0***
Wu-Hausman statistic			245.5***	155.8***
Sargan statistic			0.156	20.1***

Notes: (i) Robust standard errors clustered by app and cluster-month in parentheses.

(ii) *p<0.1; **p<0.05; ***p<0.01.

(iii) IVs: $competing\_rating_{kct-2}$, HHI_down_{jct-1}, $dev\_updates_{dct-1}$, $k \neq j$.

Table 4: Determinants of third-party apps' performance in Play Store

Dependent Variable:	$log(downloads_{jct})$			
Model:	(1)	(2)	(3)	(4)
	OLS	OLS	2SLS	2SLS
Variables				
SIM_{jct}	0.1428	0.1419	0.5160	0.6507
	(0.4246)	(0.4239)	(0.7577)	(0.5131)
TOP_SIM_{jct}	-0.7177**	-0.7598**	-2.179***	-0.9274**
	(0.3136)	(0.3137)	(0.6701)	(0.3851)
$Updates_{jct}$	0.2491***	0.2491***	4.204***	3.203***
	(0.0062)	(0.0062)	(0.7088)	(0.3990)
$First\_Party_{ct}$	0.1149*		-6.470***	
	(0.0690)		(1.240)	
$First\_Avg\_Age_{ct}$		-0.0018		-0.0464***
		(0.0012)		(0.0090)
App_price_{jct-1}	-0.0011	-0.0012	0.0048	-0.0004
	(0.0036)	(0.0036)	(0.0048)	(0.0038)
$Cum\_Ratings_{jct-1}$	0.3722***	0.3730***	0.5040***	0.4795***
	(0.0120)	(0.0120)	(0.0283)	(0.0191)
App_Rating_{jct}	0.3373***	0.3388***	0.1173*	0.1582***
	(0.0309)	(0.0310)	(0.0636)	(0.0457)
App_Age_{jct}	-0.0274***	-0.0276***	-0.0306***	-0.0217**
	(0.0090)	(0.0090)	(0.0114)	(0.0086)
Fixed-effects				
Cluster ID	Yes	Yes	Yes	Yes
Month	Yes	Yes	Yes	Yes
App	Yes	Yes	Yes	Yes
Fit statistics				
Observations	712,548	712,548	712,548	712,548
R^2	0.70696	0.70696	-0.37349	0.13693
F-statistics ($Updates_{jct}$)			123.4***	123.4***
F-statistics ($First\_Party_{ct}$)			2,022.0***	
F-statistics ($First\_Avg\_Age_{ct}$)				5,117.6***
Wu-Hausman statistic			1,107.2***	650.6***
Sargan statistic			0.2928	289.7***

Notes: (i) Robust standard errors clustered by app and cluster-month in parentheses. (ii) *p<0.1; **p<0.05; ***p<0.01. (iii) IVs: $competing\_rating_{kct-2}$, $competing\_updates_{kct-1}$, $competing\_top\_rev_{kct-2}$, $k \neq j$. (iv) The reported R^2 from 2SLS columns is negative, as 2SLS does not minimize residual variance. We instead focus on the first-stage F-statistics and over-identification tests to assess the validity of the 2SLS model.

Table 5: First-party app effects

Dependent Variable:	$\log(downloads_{jct})$			
Model:	App Store		Play Store	
	(1)	(2)	(3)	(4)
Variables				
SIM^{EX}_{jct}	0.6259*	0.5854*	1.632**	0.6740
	(0.3490)	(0.3498)	(0.8242)	(1.110)
$TOP\_SIM^{EX}_{jct}$	-0.9384***	-0.9534***	-3.876***	-8.309***
	(0.2445)	(0.2511)	(1.206)	(2.108)
$Updates_{jct}$	0.5839***	0.5967***	4.389***	5.162***
	(0.1236)	(0.1234)	(0.8029)	(1.050)
SIM_First_{ct}	3.459***		-20.56***	
	(1.012)		(5.812)	
$SIM\_First\_Like_{ct}$		3.535***		-21.55***
		(1.058)		(5.063)
App_price_{jct-1}	-0.0444***	-0.0445***	0.0045	0.0015
	(0.0151)	(0.0152)	(0.0044)	(0.0051)
$Cum\_Ratings_{jct-1}$	0.4226***	0.4210***	0.4963***	0.5276***
	(0.0096)	(0.0098)	(0.0301)	(0.0389)
App_Rating_{jct}	-0.1683***	-0.1646***	0.0499	0.0033
	(0.0255)	(0.0253)	(0.0717)	(0.0907)
App_Age_{jct}	-0.0122***	-0.0121***	-0.0316**	-0.0409*
	(0.0030)	(0.0030)	(0.0146)	(0.0216)
Fixed-effects				
Cluster ID	Yes	Yes	Yes	Yes
Month	Yes	Yes	Yes	Yes
App	Yes	Yes	Yes	Yes
Fit statistics				
Observations	546,269	546,269	682,051	682,051
R^2	0.82901	0.82738	-0.51130	-1.0860
F-statistics ($Updates_{jct}$)	295.9***	295.8***	121.1***	120.8***
F-statistics (SIM_First_{ct})	7,442.9***		706.4***	
F-statistics ($SIM\_First\_Like_{ct}$)		5,387.2***		706.4***
Wu-Hausman statistic	202.2***	217.8***	697.4***	1,024.7***
Sargan statistic	7.963***	4.1501**	143.9***	9.822***

Notes: (i) Robust standard errors clustered by app and cluster-month in parentheses. (ii) *p<0.1; **p<0.05; ***p<0.01. (iii) The instrumental variables are the same with the main analysis reported in Table 3 and 4.

Table 6: Robustness check analysis (App Store)

Dependent Variable:	$log(downloads_{jct})$			
Model:	(1)	(2)	(3)	(4)
Variables				
SIM_{jct-1}	0.4989	0.6389*		
	(0.3369)	(0.3389)		
TOP_SIM_{jct-1}	-0.5646***	-0.4024**		
	(0.2081)	(0.2050)		
SIM_{jct-2}			0.4505	0.5905*
			(0.3383)	(0.3402)
TOP_SIM_{jct-2}			-0.5335**	-0.3711*
			(0.2086)	(0.2055)
$Updates_{jct}$	0.2115***	0.1939***	0.2030***	0.1857***
	(0.0356)	(0.0366)	(0.0353)	(0.0363)
$First\_Party_{ct}$	1.230***		1.230***	
	(0.3816)		(0.3929)	
$First\_Avg\_Age_{ct}$		0.0129***		0.0125***
		(0.0040)		(0.0040)
App_price_{jct-1}	-0.0487***	-0.0483***	-0.0496***	-0.0493***
	(0.0167)	(0.0167)	(0.0174)	(0.0174)
$Cum\_Ratings_{jct-1}$	0.3979***	0.4009***	0.4035***	0.4059***
	(0.0106)	(0.0105)	(0.0112)	(0.0111)
App_Rating_{jct}	-0.1357***	-0.1374***	-0.1393***	-0.1410***
	(0.0232)	(0.0233)	(0.0239)	(0.0241)
App_Age_{jct}	-0.0142***	-0.0129***	-0.0142***	-0.0129***
	(0.0036)	(0.0039)	(0.0037)	(0.0040)
Fixed-effects				
Cluster ID	Yes	Yes	Yes	Yes
App	Yes	Yes	Yes	Yes
Month	Yes	Yes	Yes	Yes
Fit statistics				
Observations	431,346	431,346	418,544	418,544
R^2	0.86892	0.86992	0.87272	0.87363
F-statistics ($Updates_{jct}$)	$3,028.6^{***}$	$3,028.6^{***}$	$2,986.7^{***}$	$2,986.7^{***}$
F-statistics ($First\_Party_{ct}$)	$9,833.9^{***}$	$9,638.1^{***}$		
F-statistics ($First\_Avg\_Age_{ct}$)			$11,315.2^{***}$	$11,216.1^{***}$
Wu–Hausman statistic	149.6***	87.3***	136.2***	79.3***
Sargan statistic	0.178	24.9***	0.100	23.0***

Notes: (i) Robust standard errors clustered by app and cluster-month in parentheses.

(ii) *p<0.1; **p<0.05; ***p<0.01.

(iii) IVs: $competing\_rating_{kct-2}$, HHI_down_{jct-1}, $dev\_updates_{dct-1}$, $k \neq j$.

Table 7: Robustness check analysis (Play Store)

Dependent Variable:	$\log(downloads_{jct})$			
Model:	(1)	(2)	(3)	(4)
Variables				
SIM_{jct-1}	-0.4148	0.0860		
	(0.7052)	(0.4817)		
TOP_SIM_{jct-1}	-2.072***	-0.7993**		
	(0.6297)	(0.3665)		
SIM_{jct-2}			-0.8172	-0.2466
			(0.7090)	(0.4682)
TOP_SIM_{jct-2}			-1.927***	-0.8165**
			(0.6232)	(0.3622)
$Updates_{jct}$	3.676***	2.900***	3.603***	2.686***
	(0.6543)	(0.4079)	(0.6584)	(0.3963)
$First\_Party_{ct}$	-5.990***		-5.994***	
	(1.091)		(1.110)	
$First\_Avg\_Age_{ct}$		-0.0460***		-0.0443***
		(0.0087)		(0.0085)
App_price_{jct-1}	0.0041	6.37×10^{-5}	0.0037	-2.41×10^{-5}
	(0.0052)	(0.0050)	(0.0052)	(0.0051)
$Cum\_Ratings_{jct-1}$	0.4562***	0.4512***	0.4460***	0.4418***
	(0.0212)	(0.0171)	(0.0209)	(0.0172)
App_Rating_{jct}	0.1580***	0.1882***	0.1794***	0.2156***
	(0.0595)	(0.0459)	(0.0595)	(0.0451)
App_Age_{jct}	-0.0300***	-0.0211**	-0.0297***	-0.0209**
	(0.0107)	(0.0085)	(0.0109)	(0.0086)
Fixed-effects				
App	Yes	Yes	Yes	Yes
Month	Yes	Yes	Yes	Yes
cluster_id	Yes	Yes	Yes	Yes
Fit statistics				
Observations	605,609	605,609	581,661	581,661
R^2	-0.13579	0.23764	-0.08718	0.32225
F-statistics ($Updates_{jct}$)	88.2***	88.2***	86.7***	86.7***
F-statistics ($First\_Party_{ct}$)	$1,940.6$***		$1,838.4$***	
F-statistics ($First\_Avg\_Age_{ct}$)		$4,687.7$***		$4,518.6$***
Wu–Hausman statistic	949.0***	557.2***	881.0***	492.5***
Sargan statistic	9.08***	262.9***	7.89***	286.5***

Notes: (i) Robust standard errors clustered by app and cluster-month in parentheses.
(ii) *p<0.1; **p<0.05; ***p<0.01.

Table 8: Distribution of app-level autocorrelations for cosine similarity measures

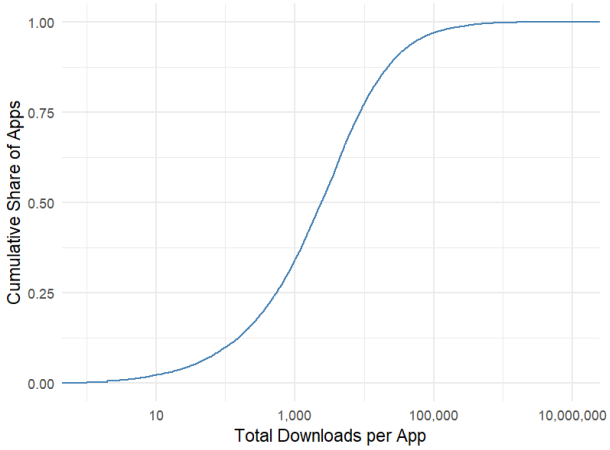
	1st Qu.	**Median**	**Mean**	**3rd Qu.**
Panel A. App Store				
SIM ρ_1	0.7186	0.8695	0.7604	0.9466
SIM ρ_2	0.5356	0.7775	0.6509	0.9043
TOP_SIM ρ_1	0.8370	0.9110	0.8440	0.9500
TOP_SIM ρ_2	0.7350	0.8440	0.7740	0.9110
Panel B. Play Store				
SIM ρ_1	0.429	0.728	0.577	0.884
SIM ρ_2	0.206	0.582	0.448	0.820
TOP_SIM ρ_1	0.649	0.840	0.717	0.934
TOP_SIM ρ_2	0.447	0.728	0.588	0.881

Notes: (i) $\rho_1 = \text{corr}(x_t, x_{t-1})$; $\rho_2 = \text{corr}(x_t, x_{t-2})$, where x indicates cosine similarity variables (SIM and TOP_SIM). (ii) Statistics computed across apps.

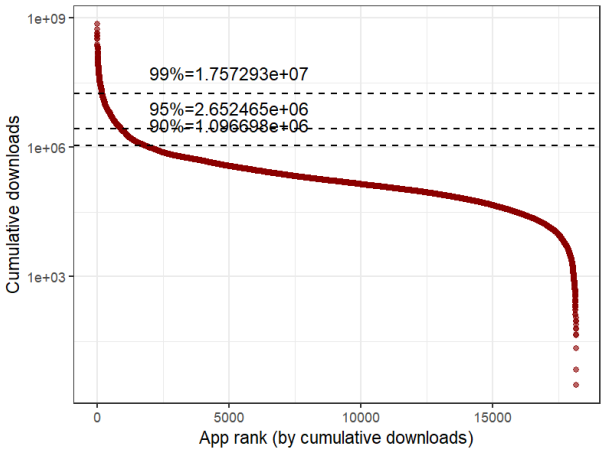
7 Figures

Figure 1: Distributions of Monthly (Cumulative) App downloads (May, 2021)

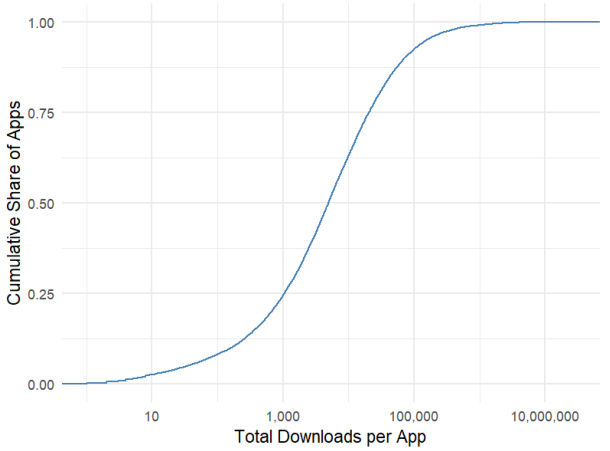
(a) CDF (App Store)



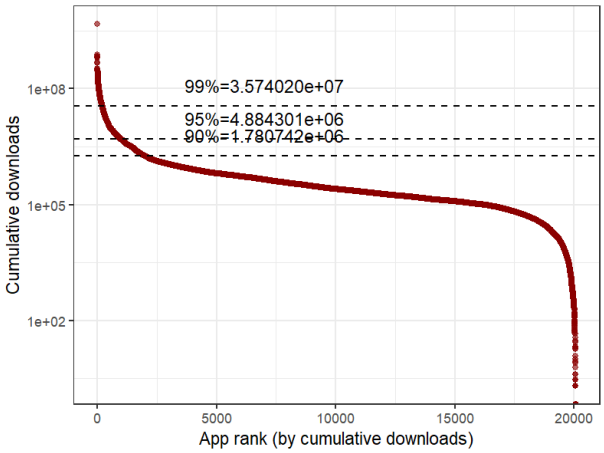
(b) Scatter Plot (App Store)



(c) CDF (Google Play)



(d) Scatter Plot (Google Play)

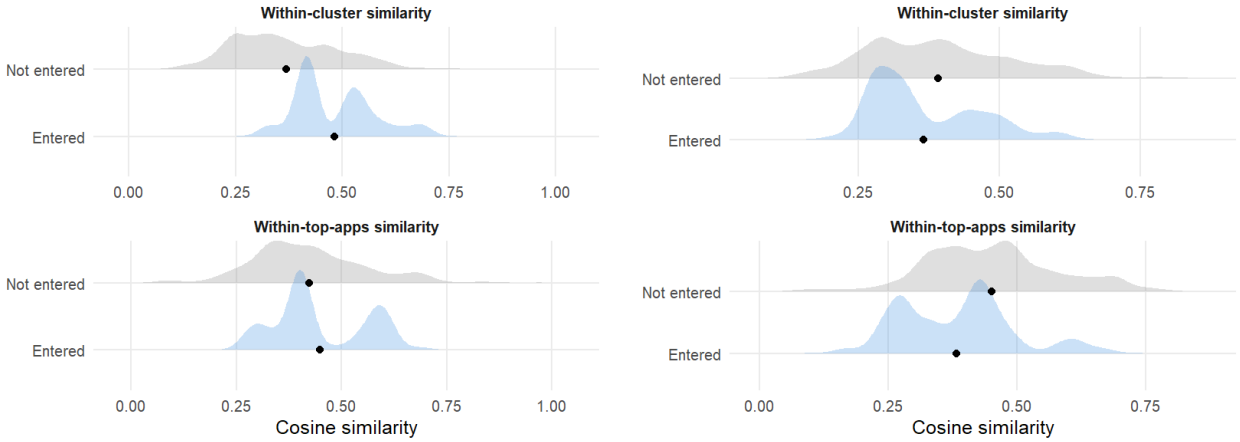


Notes: (i) Based on the panels (a) and (c), for both app stores, only a small portion (less than 10%) of apps have monthly app downloads greater than 100,000. (ii) Panels (b) and (d) both illustrate that sub-samples at the 90% cutoff correspond to 1,214 unique apps in the App Store and 1,746 unique apps in the Play Store.

Figure 2: Distributions of Cosine Similarity

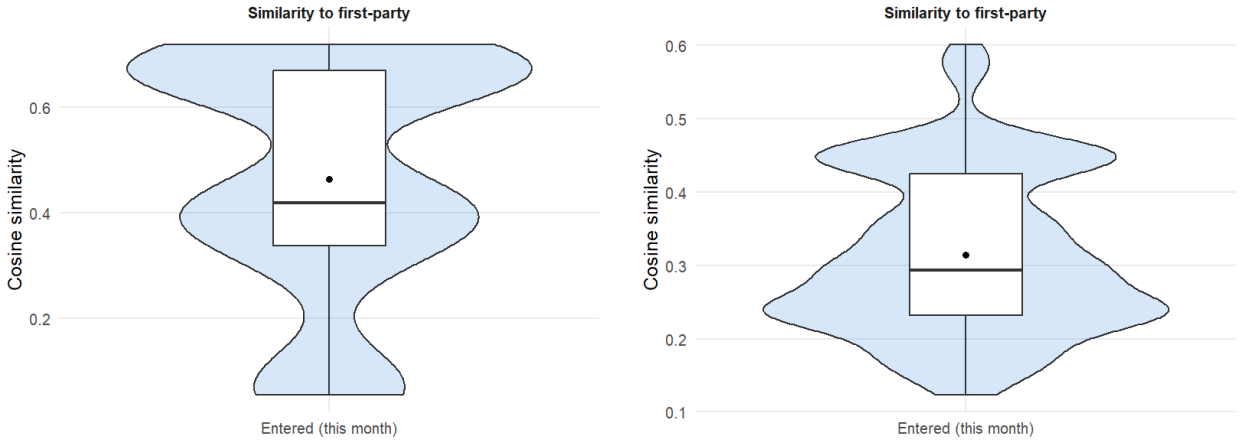
(a) App Store

(b) Google Play



(c) App Store

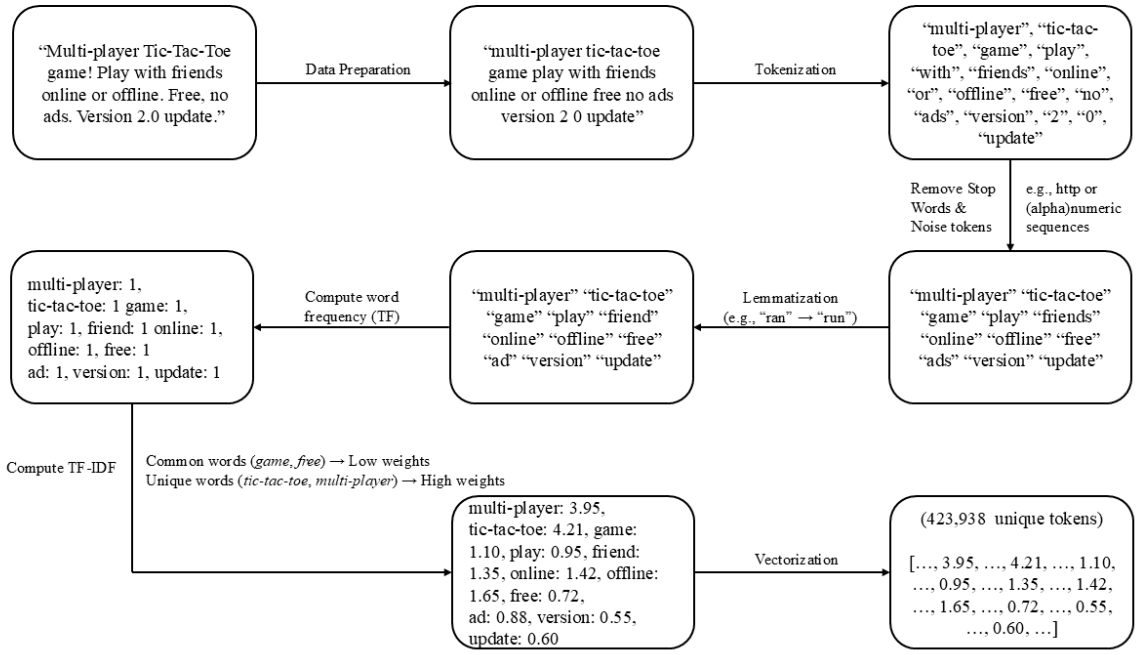
(d) Google Play



Notes: Figures (a) and (b) depict the distributions of within-cluster cosine similarity for clusters without first-party apps (upper panel) and with first-party apps (lower panel), respectively. Figures (c) and (d) present the distributions of cosine similarity between first-party apps and third-party apps within the same clusters. In all figures, black dots denote median values, while the black horizontal lines in figures (c) and (d) indicate mean values.

A Natural Language Processing

Figure A.1: NLP Pipeline



Note: This figure shows the whole process of converting the app descriptions into individual TF-IDF score tokens for uni-grams. We perform the same process for the bi-gram words as well, such as *"music player"* or *"photo editing."*

In this section, we describe the Natural Language Processing (NLP) that we used to convert app description textual data into a numeric vector of TF-IDF scores. Figure A.1 shows the pipeline of NLP. We begin by constructing a clean corpus of app descriptions. From the raw dataset, we exclude extremely short descriptions (fewer than 10 characters) as noise, and duplicate apps across app stores are removed to create a set of unique app-store observations. Preprocessing then normalizes the text by converting all characters to lowercase and unifying typographic variants such as en/em dashes (–, —) into a standard hyphen (-). Tokenization is performed using a regular expression filter that retains only letters, digits, and hyphens. This process preserves compound terms like "multi-player" while excluding punctuation and other irrelevant characters.

Next, we implement a multilingual noise-reduction step. We remove the Stopwords lists in various languages such as English, Spanish, Portuguese, Vietnamese, and so on, combined with a set of tokens that frequently arise from HTML scraping (e.g., *div*, *br*, *http*), as well as numeric tokens (e.g., *150*), alphanumeric markers (e.g., *p1*, *n20*), Unicode sym-

bols (e.g., *u2028*), and miscellaneous noises (e.g.,*ios9*, *apps*, *android*). This comprehensive stoplist ensures that the retained vocabulary is focused on semantically meaningful content. Additional filtering removes tokens that with leading or trailing hyphens, or those matching control-character patterns (e.g., $\backslash n$, $\backslash r$).

After this stage, remaining tokens are lemmatized into their dictionary forms, thereby consolidating morphological variants (e.g., *plays*, *playing* $\rightarrow$ *play*). The cleaned and lemmatized tokens form the basis for our computation of Term Frequency-Inverse Document Frequency (TF-IDF). For each app description (treated as a document), we count the frequencies of each lemmatized word. The TF-IDF weight is then assigned according to the standard formula:

$$\text{tf-idf}_{td} = f_{td} \cdot \log\left(\frac{N}{\text{df}_t}\right),$$

where f_{td} is the raw frequency of term t in document d, df_t is the number of documents in which term t appears, and N is the total number of documents. This procedure emphasizes words that are distinctive to a given app description while down-weighting those that are ubiquitous across the corpus. The resulting weighted values are stored in a sparse document–term matrix (DTM), where each row corresponds to an app ID (document) and each column corresponds to a lemmatized word. Each matrix entry contains the TF–IDF weight, producing a high-dimensional (374,214 $\times$ 423,938) but extremely sparse representation of the textual data.

A.1 Data Compression

The high dimensionality and sparsity of the dataset render the clustering computationally intensive and inefficient. To address this issue, we first employ the Truncated Singular Value Decomposition (SVD) method to reduce the dimensionality of the vector space. Truncated SVD, originally introduced by Deerwester et al. (1990), has become a widely used technique for handling high-dimensional text data and is often combined with clustering (Nur'Aini et al., 2015). The key idea is to approximate the original matrix by retaining only a small number of dimensions that capture the most important values in the data, while discarding noise and redundancy. In doing so, the retained dimensions are expressed as linear combinations of the original terms, thereby also reflecting the latent semantic structure of those terms. Although the dimensionality is dramatically reduced (even to 50-100 columns), prior studies suggest that this is sufficient to preserve the semantic structure of the word vectors, since the retained components capture the informational content of the discarded terms (Deerwester et al., 1990).

We next normalize the length of the document vector. App descriptions vary greatly in length, and relying solely on the document–term matrix (DTM) can lead to misleading similarities. Although documents of similar vector length could, in fact, contain very different content, they could be regarded as competitors. A common remedy in text analysis is to apply cosine similarity, which evaluates the angle between vectors after employing L2 normalization to rescale all document vectors to unit length. This procedure preserves the TF–IDF weights while eliminating variation due to document length, ensuring that similarity assessments correspond solely to the semantic values (Manning, 2008).

Table A.1: Apps by Clusters in App Store (Top 10 clusters out of 108)

Cluster ID	App Count	Features	Representative Apps
3	291	Photo & Video Editing and Camera Filters & Visual Design	VSCO: Photo & Video Editor, Flickr, Photoshop Express Photo Editor, Pixlr – Free Photo Editor, Instasize Photo Editor + Video, BeFunky, Picsart Photo Editor: Pic Video & Collage Maker, B612 - Best Free Camera & Photo/Video Editor, Retrica - The Original Filter Camera, HUJI FILM - Quick Shoot Camera, Candy Camera - selfie beauty camera photo editor, Sweet Selfie Camera Beauty & Filters Photo Editor, Snapseed, Perfect365: One-Tap Makeover, YouCam Nails - Manicure Salon for Custom Nail Art
16	94	Online Marketplaces & Fashion & Retail Shopping Apps	Kijiji: Buy and sell local, Poshmark: Buy & Sell Fashion, eBay: Buy sell and save on brands you love, Wallapop - Buy & Sell Nearby, Grailed - Buy & Sell Clothing, craigslist, OLX: Buy & Sell Near You with Online Classifieds, Safeway Deals & Rewards, ShopBack - The Smarter Way | Shopping & Cashback, Bed Bath & Beyond, Kroger, Stocard - Rewards Cards Wallet, Forever 21, H&M - we love fashion, Walmart - Shopping & Grocery
4	88	Language Learning & Translation and Kids Educational Games & Media	SpanishDict Translator, Dictionary.com: English Words, Dictionary - Merriam-Webster, Rosetta Stone: Learn Languages, Language Learning with Babbel, italki: Learn languages with native speakers, Cake - Learn English for Free, Speak - Practice Your English, iTranslate Converse, Translate Photo+ Scan Camera, Lingokids - playlearning, HOMER Learn & Grow, Duolingo ABC - Learn to Read, BabyTV - Baby & Toddler Videos, Google Family Link for parents

Table A.1: Apps by Clusters in App Store (Top 10 clusters out of 108) (continued)

Cluster ID	App Count	Features	Representative Apps
7	69	Music Streaming & Audio Ecosystem	Smule, WeSing, Karaoke - Sing Unlimited Songs, Spotify, Deezer, JOOX Music, SoundCloud, TREBEL, Synthesia, Simply Piano by JoyTunes, Flowkey, Guitar Play & Learn, Music Maker JAM, edjing Mix, Relax Melodies
12	62	Fitness & Health & Lifestyle Management	Pacer Pedometer, Pedometer++, Strava, adidas Running App, Sweatcoin, Calorie Counter by FatSecret, MyNetDiary, Zero, Fastic, Period Tracker by GP Apps, JEFIT Workout Planner, 8fit, Gymshark Training, Six Pack in 30 Days, Garmin Connect
5	51	Streaming & Live TV Apps	Paramount+, CBS News, NBC News, Netflix, Peacock TV, Discovery+, Pluto TV, Spectrum TV, Freeform TV, Vudu, Telemundo, Eros Now, SiriusXM, Apple Podcasts, Streamlabs
11	49	Navigation & Mobility Apps	Google Maps, Waze, MapQuest, Moovit, 2GIS, Yandex Maps, Roadtrippers, CityMaps2Go, GasBuddy, BlaBlaCar, Lime, Yandex Go, Cabify, Cars.com, PayByPhone Parking
23	48	Productivity: Scanning, Storage & Office Tools	MediaFire, WinZip, 4shared, Canon PRINT Inkjet/SELPHY, HP Samsung Mobile Print, Adobe Scan, CamScanner HD, Genius Scan, Scanner App: PDF Document Scan, PhotoScan by Google Photos, OfficeSuite, PDF Reader, Adobe Fill & Sign, MarginNote, MindNode
10	36	Investment & Personal Finance Apps	thinkorswim, Robinhood, Stash, Wealthsimple Trade, Halifax Mobile Banking, Monzo, Starling Bank, U.S. Bank, Capital One CreditWise, Netspend, Varo, MoneyLion, Truebill, Affirm, SumUp

Table A.1: Apps by Clusters in App Store (Top 10 clusters out of 108) (continued)

Cluster ID	App Count	Features	Representative Apps
6	28	Airlines & Flight Booking	TripCase, Alaska Airlines, Vueling Airlines, Eurowings, British Airways, Norwegian Travel Assistant, Spirit Airlines, ANA, airasia, Qantas Airways, Skyscanner, Opodo, Last Minute, Flightradar24, FlightStats

Table A.2: Apps by Clusters in Play Store (Top 10 clusters out of 107)

Cluster ID	App Count	Features	Representative Apps
1	220	Content Creation & Sharing Tools (Photo/Video/Story)	Snapseed, VSCO, Picsart Photo Editor, Pixlr, Photoshop Express, Retrica, Candy Camera, B612, Remini, Instasize, Boomerang from Instagram, VN Video Editor Maker VlogNow, VideoShow Video Editor, Triller, StoryArt
8	165	Shopping & Payment Apps	eBay, Amazon, Walmart, Coupang, Shopee, JD Central, Club Factory, Wish, GOAT, H&M, Nike, Forever 21, McDonald's, PayPal, Google Pay
2	115	E-books, News Media, and Navigation Tools	Kobo Books, Libby by OverDrive, The New York Times, The Wall Street Journal., CNN, Al Jazeera English, Google News, SmartNews, NewsBreak, Wikipedia, WordPress, Mailchimp, Gmail, Google Keep, Google Maps
4	100	Music Streaming, Learning & Creation	Smule, WeSing, Simply Piano by JoyTunes, Synthesia, Guitar - Real games & lessons, Fender Guitar Tuner, Deezer, Spotify, YouTube Music, Amazon Music, SoundCloud, JOOX Music, SiriusXM, Audible, edjing Mix
34	100	Educational & Entertainment Apps for Children	Baby Panda Care, Baby Panda's Supermarket, Baby Panda's Math Adventure, Baby Panda's Treasure Island, Baby Panda's Emergency Tips, Baby Panda's Kids Safety, Baby Phone for toddlers, Preschool Games for Toddler, MentalUP, Coloring Book for Kids, Princess Coloring Book Glitter, Talking Angela, Talking Ginger, Masha and the Bear Educational Games, Smolsies My Cute Pet House

Table A.2: Apps by Clusters in Play Store (Top 10 clusters out of 107) (continued)

Cluster ID	App Count	Features	Representative Apps
40	92	Mobile Personalization & Security Tools	ZEDGE Wallpapers & Ringtones, GO Launcher, Apex Launcher, Smart Launcher 6, Action Launcher: Pixel Edition, Super S10 Launcher Galaxy S10, XOS Launcher 2022, 3D Wallpapers, 4K AMOLED Wallpapers, Anime Wallpaper, GRUBL 4D Live Wallpapers, AppLock, AppLock - Fingerprint Password, Gallery Lock, Calculator Vault
16	84	Communication Hub: Calls, Recording & Messaging	Phone by Google, WhatsApp Messenger, Viber, Discord, Microsoft Teams, Zoom, Webex Meetings, GoToMeeting, CallApp: Caller ID & Recording, Whowho, Line2 - Second Phone Number, 2nd Phone Number App, Automatic Call Recorder - ACR, Mobizen Screen Recorder, Voice Recorder
14	77	File Management & Document Tools	Files by Google, MX Player, MediaFire, 4shared, SHARE ALL (File Transfer), Zender, HP Print Service Plugin, Canon PRINT Inkjet/SELPHY, Samsung Mobile Print, Adobe Scan, Genius Scan, OfficeSuite, PDF Reader, WinZip, EaseUS MobiSaver
13	76	Mobile Security & Utility Tools	When I Work Staff Scheduling, QuickBooks Time Tracking, Norton 360: Mobile Security, AVG AntiVirus & Security, Kaspersky Security & VPN, Avast Cleanup – Phone Cleaner, CCleaner – Phone Cleaner, Nox Cleaner, Greenify, Samsung Pay, Samsung Internet Browser, Samsung Smart Switch Mobile, Microsoft Power BI, Android Auto for phone screens, Multiple Accounts: Dual Accounts & Parallel Space

Table A.2: Apps by Clusters in Play Store (Top 10 clusters out of 107) (continued)

Cluster ID	App Count	Features	Representative Apps
3	63	Streaming & Smart TV Ecosystem	Netflix, Disney+, Paramount+, Peacock TV, Spectrum TV, DIRECTV STREAM, Pluto TV, Vudu, Freeform TV, FXNOW, Telemundo, Eros Now, Univision Now, Roku Remote (Rokie), TeamViewer Host

B Additional Tables

Table B.1: First Party Apps (Apple)

	App	Cluster ID	Leading Apps
1	Find My iPhone	57	0
2	iTunes Connect	17	0
3	FileMaker Go 13	27	0
4	Clips	77	0
5	Apple Heart Study	27	0
6	Indoor Survey	27	0
7	Apple Store	16	1
8	Dark Sky Weather	21	1
9	Apple Podcasts	5	1
10	Apple Developer	68	1

Table B.2: First Party Apps (Google)

	App	Cluster ID	Leadging Apps
1	Google AdSense	61	0
2	Photowall for Chromecast	19	0
3	Voice Search	19	0
4	Google Now Launcher	37	0
5	Cardboard Camera	1	0
6	Accessibility Scanner	24	0
7	Daydream Keyboard	48	0
8	Trusted Contacts	57	0
9	Science Journal	84	0
10	Just a Line - Draw in AR	93	0
11	Google I/O 2019	100	0
12	Recorder	16	0
13	Android System WebView Beta	64	0
14	Google Earth	2	1
15	Hangouts	16	1
16	Snapseed	1	1
17	Google Authenticator	24	1
18	Google News - Top world & local news headlines	2	1
19	Chrome Beta	9	1
20	Google Find My Device	2	1
21	Cardboard	24	1
22	YouTube Studio	1	1
23	Google Keep - Notes and Lists	2	1
24	Gmail - Email by Google	2	1
25	Google Chrome	9	1
26	Google Play Music	4	1
27	Android Accessibility Suite	13	1
28	Google Fit: Activity Tracking	6	1
29	Phone by Google	16	1
30	Chrome Canary (Unstable)	9	1
31	PhotoScan by Google Photos	14	1
32	Grasshopper: Learn to Code	24	1
33	Fabby Look: hair color changer	34	1
34	Jamboard	14	1
35	Files by Google	14	1
36	Google Family Link for parents	2	1
37	Sound Amplifier	13	1
38	Google Maps	2	1
39	Google Pay for Business	8	1
40	YouTube Music	4	1
41	Read Along: Google	2	1
42	Android Auto for phone screens	13	1

Table B.3: Correlations across variables (App Store)

		1	2	3	4	5	6	7	8	9	10
1	App downloads	1	0.013	-0.017	0.040	0.063	0.301	0.402	-0.106	0.155	-0.336
2	Similarity	0.013	1	0.750	0.233	0.188	0.009	-0.009	-0.018	0.009	-0.029
3	Similarity (leading apps)	-0.017	0.750	1	0.192	0.124	-0.028	-0.047	-0.021	-0.035	-0.041
4	Similarity (first-party)	0.040	0.233	0.192	1	0.916	0.073	0.041	-0.014	0.007	0.052
5	First-party presence	0.063	0.188	0.124	0.916	1	0.090	0.053	-0.016	0.005	0.043
6	App updates	0.301	0.009	-0.028	0.073	0.090	1	0.205	-0.019	0.158	-0.107
7	Cumulative ratings	0.402	-0.009	-0.047	0.041	0.053	0.205	1	-0.038	0.355	0.204
8	App price ($)	-0.106	-0.018	-0.021	-0.014	-0.016	-0.019	-0.038	1	0.023	0.035
9	Average star rating	0.155	0.009	-0.035	0.007	0.005	0.158	0.355	0.023	1	-0.133
10	App age (months)	-0.336	-0.029	-0.041	0.052	0.043	-0.107	0.204	0.035	-0.133	1

Table B.4: Correlations across variables (Play Store)

		1	2	3	4	5	6	7	8	9	10
1	App downloads	1	-0.017	-0.044	0.101	0.151	0.233	0.343	-0.117	0.120	-0.229
2	Similarity	-0.017	1	0.728	0.090	-0.031	0.036	-0.015	-0.014	0.076	-0.008
3	Similarity (leading apps)	-0.044	0.728	1	-0.052	-0.219	0.030	-0.069	-0.013	-0.001	0.011
4	Similarity (first-party)	0.101	0.090	-0.052	1	0.820	0.066	0.144	-0.009	0.081	-0.017
5	First-party presence	0.151	-0.031	-0.219	0.820	1	0.070	0.187	-0.010	0.076	-0.031
6	App updates	0.233	0.036	0.030	0.066	0.070	1	0.144	-0.017	0.089	-0.022
7	Cumulative ratings (lag 1)	0.343	-0.015	-0.069	0.144	0.187	0.144	1	-0.091	0.207	0.346
8	App price ($)	-0.117	-0.014	-0.013	-0.009	-0.010	-0.017	-0.091	1	0.012	0.067
9	Average star rating	0.120	0.076	-0.001	0.081	0.076	0.089	0.207	0.012	1	-0.061
10	App age (months)	-0.229	-0.008	0.011	-0.017	-0.031	-0.022	0.346	0.067	-0.061	1